Assignment 2

Well Done on implementing the backend of your application. This is just the beginning.

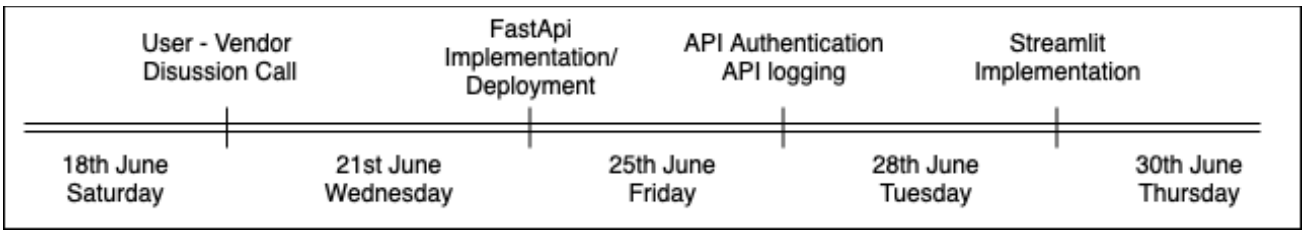
In Assignment 2 we will try to go one step ahead and deploy your backend and make a beautiful webapp.

Things you need to do:

1) ***API vendor-user Discussion call***
2) ***FAST Api Implementation/Deployment***
3) ***API Authentication***
4) ***Streamlit Implementation***
5) ***API Logging***

Interesting things to do this week! 😅

Suggested Timeline of things: (This is a suggested timeline; you can work on your speed just make sure you complete the assignment)



Note:

1) You need to create a repository inside your organization as ***Assignment2***

2) Make sure there are no errors in your python files when you submit it.

3) Have a requirement.txt file in your repository so we can install packages which you have used.

API vendor-user Discussion call

In this module you will be *an API user* and will have a discussion call with your *API vendors* on the functions which are created by the vendor.

As an *API user* you are expected to know what the function made by your vendor does, what are the inputs and what are the outputs, what types are allowed etc.

API Vendor	***API User***
Team 1	***Team 2***
Team 2	***Team 3***
Team 3	***Team 4***
Team 4	***Team 1***

For ex: Team 2 should know about every function team 1 has made (Vendor) likewise for all teams.

Fast API implementation/Deployment

Implementation:

For this module you are the API vendor, once you and your user agree upon the functionality of the functions you need to implement the functions with FAST Api.

(P.S: You have already created the code for your functions in assignment 1, in this module you need to wrap those functions with Fast Api.

Fast API Tutorial:

https://www.youtube.com/watch?v=-ykeT6kk4bk&t=143s

https://www.youtube.com/watch?v=1zMQBe0l1bM

Why is this required? The functions which you created can only be executed on your local machines, FAST Api will allow us to share those functions with your user group and with the world.

Submission: Push your codebase of Fast API on GitHub (Assignment 2 Repository)

Deployment:

When you are ready with FAST Api code base, you need to deploy this on a server. When you run Fast Api locally you can fire up uvicorn server but that gets executed on local host and that link is not shareable. In this module you will need to deploy the code base on any of the following services.

1) Heroku: https://www.tutlinks.com/create-and-deploy-fastapi-app-to-heroku/
2) Google App Engine: https://www.tutlinks.com/deploy-fastapi-app-on-google-cloud-platform/
3) AWS
4) You can find your own as well

Tutorial on how to Deploy a basic API on Heroku will be taken in TA hours. But the concept remains the same for the rest of all services as well. So, you can choose according to your use case.

Submission: The link which will be generated after the deployment.

Fast API Authentication

In this module you will be securing the FASTApi with JWT tokens.

To do: You need to create 5 users in your API Database

1) A username, password for your team
2) 2 usernames, passwords for your users (Another team who will build a streamlit on your API)
3) A username, password for Prof
4) A username, password for Parth!

Do not hard code the login page, use JWT tokens and bearers to do so, store the hashing in a database and authenticate users with that hashings!

https://fastapi.tiangolo.com/tutorial/security/oauth2-jwt/

Submission: In Streamlit application you should have a login page where you authenticate the users!

Streamlit:

In this module you will build a web application to use the deployed the API.

Note: It should be a Multi-Page Application

You will have 3 Pages in your streamlit:

1) Authentication Page
2) Functions Page
3) API logging Page/Analytics of your API

Few Examples:

1) https://github.com/deepchecks/checks-demo
2) https://streamlit.io/gallery

Submission: You need to deploy your streamlit on cloud so that its accessible to everyone.

API Logging/Analytics:

In this module you will be implementing your user logging for the API calls.

This means that you should save everything a user does when they call your API.

Ex:

1) Number of hits a particular end point got.
2) Number of bad requests.
3) User wise error code distribution
4) Etc.

You can create a database and save the entire journey of a user in that database, or you can choose your own method.

Submission: Create a dashboard using python (Plotly, etc) and embed the dashboard in streamlit.

Extra Notes:

1) Create a YouTube video where you explain how to use your streamlit application.
2) YouTube video submission date is 2nd July.

For the week of Holiday: 2nd July – 9th July)

Every Team needs to test the API/ API Documentation/ Streamlit and its functionality of other team as below.

Team 2 will make a test report for Team 1 (Team 2 is the user)

Team 3 will make a test report for Team 2 (Team 3 is the user)

Team 4 will make a test report for Team 3 (Team 4 is the user)

Team 1 will make a test report for Team 4 (Team 1 is the user)

Test report should be a detail document testing all the components for your API developer.

This will be rechecked by the Prof and TA so if in any major code issue found in API and not got caught in testing document, will make lose marks for the team who is testing it!

\